being dependent upon a rejected base claim but would be allowable if rewritten in independent form. Applicants respectfully traverse these rejections, and request reconsideration and allowance of the pending claims in view of the following arguments. Of the rejected claims, claims 1, 13, 25, and 37 are independent.

The invention disclosed in the application relates to a method and apparatus for directing and assisting a user through procedures of a program required to perform various tasks on a complex software system (specification, p. 1, lines 12-14). Independent claims 1, 13, 25, and 37 in the present application recite a program procedure which assists a user through a procedure on a computer, in particular, a software installation, migration, or updating procedure. The user sets or changes parameters only in a parameter input window.

The term "program procedure" clearly refers to software installation and updating, when term is viewed in the context of the specification, rather than in a vacuum. At page 8 of the specification, beginning at line 11, describing a preferred embodiment of the invention, there is a discussion of the large number of parameters that may be involved. At page 14, beginning at line 11, there is a discussion of how parameters may be made available to the user to either change or not, as the user wishes.

Thus, when the claims are read not in a vacuum, but in light of the specification, the context of "program procedure" is not nearly so broad as the Examiner seems to be asserting. Looking at the prior art the Examiner has cited, and the way in which the Examiner has read the rejected claims on that prior art, it appears clear that the Examiner has construed the claims so broadly as to be inconsistent with the specification.

Benton relates to monitoring devices on an automation system through graphical representations. This has nothing to do with installation, migration, or updating, on which the proper interpretation of "program procedure" reads. Likewise, Paterson, which relates to a method of modifying parameters in a simulation model, has nothing to do with these functions. Neither does Massaro, which relates to selecting function complexity for data processing systems. Graphically representing devices on an automation system, selecting parameters for a system model, and selecting function complexity for a data processing system have nothing to do with assisting a user in the installation or updating of software.

Looking more closely at the prior art, Benton is directed to a system which allows a user to create software representations of devices, such as security and fire control structures, in an industrial environment or control structure. (Benton col. 1, lines 20-29). Benton also relates to a system that creates graphical representations of device parameters. (Benton col. 9, lines 12-15). A user controls and monitors the devices through the graphical representation.

The Examiner relied on Benton for teaching a method of leading a user through a program procedure on a computer by displaying a window containing parameter information. The Examiner is correct that Benton teaches a method of leading a user through a procedure on a computer, but the method taught in Benton is not a software program installation, migration, or updating procedure. The primary purpose of Benton is the control and monitoring of control structures through an automation system, which includes software. In addition, Benton teaches a method of remedying the problem of requiring numerous points for graphical representations by removing the requirement of setting up logical parameters for each graphical symbol. Benton does not assist a user in the installation, migration, or updating of software as required by the claimed invention, therefore Benton does not meet the requirements of independent claims 1, 13, 25, 37 and their dependent claims.

Paterson teaches a method of representing, accessing, inputting and monitoring parameters of a simulation model. (Paterson abstract). Simulation modeling is commonly used to model systems to perform "what-if" analyses to optimize system performance and to identify problems with systems. (Paterson col. 1, lines 13-16).

Initially, Applicants would like to respond to the Examiner's allegation that Paterson teaches or suggests the changing of parameters as claimed in the present application. The Examiner has referred to the Paterson abstract to support this position, and to contradict Applicants' assertion, made in the last response, that Paterson only teaches, at most, the changing of a graphical representation of a parameter, as does Benton.

The Examiner is correct that the Paterson Abstract (and also, in some places, the specification) refer to the changing of object parameters. However, as described in Paterson, this is done by changing GUI representations of the object parameters. *See*, *e.g.*, Fig. 18 (steps 354, 356, 358 – identifying parameters with layer panel(s) which are within a GUI; step 362, inputting parameter values via a GUI) and accompanying description at col. 17, lines 15-53.

As can be seen from the foregoing, the Examiner appears to have construed the "object parameter" language in a vacuum, and with hindsight reconstruction based on the teachings of the present application. Such an approach is improper, as a matter of law.

The Examiner also relied on Paterson as teaching the limitation of transferring the user from the window to a parameter input window associated with one of the parameters selected by the user to be set or changed, wherein the user provides information in the parameter input window to set or change the value of the parameter. The Examiner further relied on Paterson as teaching the limitation of preventing the user from selecting to set or change a value of a parameter for at least one of the parameters.

The Examiner is correct that Paterson teaches a method of transferring the user to a parameter input window associated with one of the parameters selected by the user, but Paterson teaches this transferring in the context of simulation modeling, and not in the context of software installation, migration, or updating. At no point does Paterson teach or suggest the transferring of a user to a parameter input window associated with a parameter to be changed in the context of a software installation, migration, or updating procedure.

The Examiner also argues that Paterson teaches preventing the user from changing a parameter with the CANCEL button and a baseline value that is specified by the original builder of the simulation model. The Examiner agreed with our previous argument that the CANCEL button is not the same as preventing the user from setting or changing a parameter. Instead, the Examiner now argues that the baseline value is specified by the original builder of a simulation model in which the relevant parameter is utilized and can not be modified directly by a user.

As stated above, Paterson teaches a method of providing access to parameters in a simulation model. Patterson does not teach or suggest a method of assisting a user in installing, migrating, or updating software, nor does it teach or suggest that its method of parameter modification can be performed in a context other than simulation modeling.

Benton and Paterson combined fail to meet the requirements of the independent claims of the present application because the references do not pertain to a method of assisting a user in the installation, migration, or updating of software. Even assuming *arguendo* that Benton teaches a method of leading a user through a procedure, it is only in the context of an industrial environment. Although Paterson teaches a method of changing parameters, it is not in the context of assisting a user in a software installation, migration, or updating procedure, but merely for simulation modeling. An artisan of ordinary skill would not have combined Benton with Paterson. Even if such an artisan would have combined the references, the combination fails to meet the requirements of the independent claims. Therefore, Applicants submit that independent claims 1, 13, 25, 37 and their dependencies are patentable.

As noted earlier, it may be that the Examiner is construing the "program procedure" as any type of procedure which lists a particular sequence of events, but properly interpreted, the term "program procedure" means the installation, migration, or updating of software. Applicants submit that the Examiner's construction is too broad in light of Applicants' specification, which discusses the "program procedure" as assisting a user through installing, migrating, or updating a program. (Abstract). See, for example, (description at page 2, lines 16-19).

Examiner relies on Massaro for teaching that the user is given two interaction path options each having a different level of complexity, which is lacking in Benton and Paterson. As stated above, Benton and Paterson fail to meet the requirements as claimed by the Examiner and Massaro does not supply any of the deficiencies of Benton and Paterson.

Massaro relates to an improved data processing system and to an enhanced user interface for utilization with a data processing system. Massaro teaches multiple user interfaces are established for selected functions within a multiple function application and stored within a data processing system. Each of the multiple user interfaces preferably has a different level of complexity. (Massaro abstract).

Massaro does not teach or suggest that different levels of complexity can be used in a context other than a data processing system. Massaro is merely trying to improve upon current database processing systems by determining the level of assistance a user will need with each function of the database. The amount of assistance required will depend on the level of the user's skill with data entry or data manipulation. (Massaro col. 6, lines 42-45). The options in Massaro do not assist a user in the installation or updating of software, but merely assist a user in handling data.

For these reasons, even if one of ordinary skill in the art could have combined the teachings of Benton, Paterson, and Massaro, the resulting combination would not result in the invention recited in independent claims 1, 13, 25, and 37, nor in any of the dependent claims. Accordingly, Applicants submit that independent claims 1, 13, 25, and 37, and their respective dependencies, are patentable.

The Examiner has objected to claims 7-10, 12, 19-22, 24, 31-34, 36, 43-46 and 48 as being dependent upon a rejected base claim. At this time, Applicants respectfully decline the Examiner's invitation to rewrite these claims in independent form since they depend from patentable claims, as discussed above.

For the above-outlined reasons, Applicants submit that independent claims 1, 13, 25, and 37, and their respective dependencies, are therefore patentable. Accordingly, reconsideration and allowance of the above claims is respectfully and earnestly requested.

The Examiner's rejections having been overcome, Applicants submit that the subject application is in condition for allowance. The Examiner is respectfully requested to contact the undersigned at the telephone number listed below to discuss other changes deemed necessary. Applicants hereby petition for any extension of time which may be required to maintain the pendency of this case, and any required fee for such extension is to be charged to Deposit Account No. 19-4880.

Respectfully submitted,

Frank L. Bernstein
Registration No. 31,484

SUGHRUE, MION, ZINN,
MACPEAK & SEAS, PLLC
Tel: (650) 325-5800

23493
PATENT TRADEMARK OFFICE

Date: August 1, 2001

Certificate of Mailing

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to:

Assistant Commissioner for Patents
Washington, D.C. 20231

Date: August 1, 2001

Signed:
Thea K. Wagner